AMENDMENTS TO THE CLAIMS

1. (Currently amended) An article comprising a machine-accessible medium having stored thereon instructions that, when executed by a machine, cause the machine to:

obtain, from a performance monitor, performance data for a memory heap having a plurality of memory regions;

based on the performance data, determine if at least one of the plurality memory regions is a delinquent region, wherein said determining includes instructions that when executed cause the machine to count the number of occurrences of the performance data and to compare the count of the number of occurrences of the performance data to a threshold value, wherein if the count has reached the threshold value a delinquent region is determined to exist, wherein the performance data represents at least one memory performance event; ~~and~~

in response to a determination that at least one of the plurality of memory regions is a delinquent region, execute a memory management routine to optimize that region of the memory heap by executing a garbage collection routine on at least one delinquent region, the garbage collection routine re-arranging the plurality of memory regions stored in the memory heap to optimize the memory heap; and

execute a secondary memory management routine on at least one non-delinquent region, wherein the secondary memory management routine is different than the memory management routine.

2. (Canceled)

3. (Original) The article of claim 1, wherein the performance data is selected from the group consisting of cache misses, translation lookaside buffer misses, branch mis-predicts, stalls due to data dependency, and data cache write-back.

4. (Previously presented) The article of claim 1, wherein the performance monitor is a Performance Monitoring Unit (PMU) of a central processor for the machine.

5. (Canceled)

6. (Canceled)

7. (Currently amended) The article of claim [[6]] 1, wherein the garbage collection routine is selected from the group consisting of reference counting collection, copy collection, generational collection, mark-sweep collection, beltway collection, oldest first collection, slide compaction or a hybrid collection.

8. (Original) The article of claim 1, having further instructions that, when executed by the machine, cause the machine to:

establish a size granularity of the memory region prior to obtaining the performance data for the memory region.

9. (Canceled)

10. (Canceled)

11. (Currently amended) The article of claim [[10]] 1, having further instructions that when executed by the machine, cause the machine to:

determine if a sufficient number of data samples have been taken, before comparing the count to the threshold value.

12. (Currently amended) The article of claim [[10]] 1, having further instructions that, when executed by the machine, cause the machine to:

in response to a determination that an additional data sample is to be taken, collect the additional data sample from the memory heap.

13. (Original) The article of claim 1, having further instructions that, when executed by the machine, cause the machine to:

block the delinquent region from memory storage.

14. (Currently amended) A method comprising:

within a central processor for a machine, identifying load miss memory addresses from a memory heap including a plurality of memory regions;

maintaining a frequency count for the identified load miss memory addresses;

determining if any of the plurality of memory regions ~~include~~ have a threshold value of load miss memory addresses; and

optimizing the memory heap in response to a determination that at least one of the plurality of memory regions includes a threshold value of load miss memory addresses, wherein optimizing the memory heap comprise performing a garbage collection on at least one of the memory regions including the threshold value of load miss memory addresses.

15. (Original) The method of claim 14, wherein optimizing the memory heap comprises blocking the memory regions including the threshold value of load miss memory addresses.

16. (Canceled)

17. (Currently amended) The method of claim [[16]] 1, wherein the garbage collection optimization is selected from the group consisting of the group consisting of reference counting collection, copy collection, generational collection, mark-sweep collection, beltway collection, oldest first collection, slide compaction or a hybrid collection.

18. (Original) The method of claim 14, further comprising:

performing a first memory management routine on at least one memory region including the threshold value of load miss memory addresses; and

performing a second memory management routine, different than the first memory management routine, on at least one memory region that does not include the threshold value of load miss memory addresses.

19. (Currently amended) A system comprising:

hardware to monitor performance of a memory heap and to compile performance data on memory regions within the memory heap, wherein the hardware is able to determine if any of the memory regions are delinquent regions based on the compiled performance data and wherein the hardware has a memory manager for optimizing the delinquent regions by re-arranging memory regions in the memory heap in response to a determination of at least one delinquent ~~member~~ memory region, wherein a delinquent memory region is one that has at least a threshold number of occurrences of a memory performance event, and a non-delinquent region is one that has less than a threshold number of occurrences of a memory performance event; and

[[a]] the memory manager in the form of a garbage collector for optimizing the delinquent regions using a first garbage collection routine and for optimizing non-delinquent regions using a second garbage collection routine different than the first garbage collection routine.

20. (Currently amended) The system of claim 19, wherein the hardware comprises a performance monitoring unit~~, and wherein the memory manager is a garbage collector~~.

21. (Currently amended) The system of claim [[20]] 19, wherein the garbage collector executes a garbage collection optimization selected from the group consisting of reference counting collection, copy collection, generational collection, mark-sweep collection, beltway collection, oldest first collection, slide compaction or a hybrid collection.